



Metadatadrevet publisering på multiplattformssystemer

Hans-Arne Hånes

Elektronisk systemdesign og innovasjon

Innlevert: juni 2014

Hovedveileder: Andrew Perkis, IET

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Sammendrag

Denne oppgaven belyser hvordan beskrivende informasjon om et multimedialt innhold kan benyttes som en ressurs for å danne et nyskapende system for interaksjon mellom en redaktør og en bruker. Ved å tillate en tilbakekobling fra en sluttbruker til et intelligent ledd i form av en redaktør, åpner det seg mange muligheter for interaksjon og kommunikasjon som kan gi innsikt i bruk av multimedialt innhold. For redaktøren kan det være interessant å kunne styre presentasjonen av innholdet hos sluttbrukeren. Ved å benytte tilbakekoblingen til redaktøren kan det også være interessant å samle inn informasjon i en brukerlogg fra sluttbrukeren. Redaktøren er i denne oppgaven en person med forskningsmotiv knyttet til et multimedialt innhold. Konseptet som er utgangspunktet for denne oppgaveidéen er UMA. UMA (“Universal Multimedia Access”) er et samlet begrepet for et system som sørger for optimal enhetsuavhengig multimediamkonsumering over et nettverk. I dagens systemer som fasiliterer UMA er det flere som benytter seg av metadata på forskjellige måter for å berike opplevelsen. Dette kan være i form av å gi semantiske beskrivelser som gjør tilgjengelighet bedre, eller innholdsanbefalinger som er relevant basert på brukerhistorikk.

Resultatet av forskningsarbeidet ble et demo-system bestående av en redaktørapplikasjon utviklet for PC, og en sluttbrukerapplikasjon utviklet for nettbrett som kommuniserer trådløst med hverandre gjennom en tjener. Systemet tilfredstiller i stor grad de på forhånd stilte krav og forventninger, der det viktigste målet var kommunikasjon mellom redaktør og sluttbruker. Det

ble gjennomført en brukertest som videre dokumenterer at et slikt system har nytteverdi for en redaktør med forskningsmotiv knyttet til materialet sitt. Av videre utvikling er det interessant å forske på hvor gode og troverdige metadata man får generert fra et slikt system. Med dagens smarte enheter som har sensorteknologi for bevegelse, kan et annet felt for utvikling være å åpne for å kunne sende passive tilbakemeldinger til redaktøren i form av bevegelse eller lignende.

Abstract

This thesis looks at how descriptive information about multimedia content can be used as a resource to create an innovative system for interaction between an editor and a user. Permitting a loopback from an end-user to an intelligence in the form of an editor, opens up a great deal of opportunities for interaction and communication. In this thesis two main functionalities are explored. First, enabling the editor to control the presentation of content at the end-user. Second, the loopback provides the editor with information from the end-user gathered in a user-log. This may give the editor valuable insight into the end-user's perception of the multimedia content. The editor can in this thesis be viewed as a person with motives of research connected to the multimedia content. The concept of which this idea is based on, is UMA. UMA("Universal Multimedia Access") is a term used to describe a system that provides optimal ubiquitous multimedia consumption through a network. In today's systems, which facilitates UMA, there are many examples of ways to enrich the user experience by the use of metadata. This can be in terms of giving semantic descriptions which leads to better availability, or content recommendations which is relevant to a user based on user history.

The result of the research work is a demo-system consisting of an editor application developed for PC, and an end-user application developed for tablet, which communicates wirelessly with each other through a server. The system satisfies most of the requirements and expectations which were set in

advance. The most important requirement was the communication between editor and end-user. A user test was carried out to further document that such a system is of value to an editor. As a further development it can be interesting to investigate how good and credible the metadata generated in this system is. With today's smart devices containing sensor technology, another field of development can be to look at the possibility of sending passive feedback such as movement to the editor.

Forord

Denne oppgaven ble utført siste året av masterstudiet i Elektronikk ved Norges teknisknaturvitenskapelige universitet (NTNU) i Trondheim, med retning signalbehandling, akustikk og multimedia. Oppgaven ble påbegynt våren 2014.

Jeg vil gjerne takke min hovedveileder Professor Andrew Perkis og ressursperson Andreas Bergsland for gode innspill og hjelp underveis. Når det gjelder teknisk hjelp, har jeg satt stor pris på den hjelpen jeg har fått av Jordi Puig til programmering i Max. Det har også vært til stor hjelp at venner og bekjente har stilt opp på brukertest.

Hans-Arne Hånes

Trondheim, juni 2014

Innhold

1	Introduksjon	1
1.1	Mål for oppgaven	2
1.2	Oppgavens struktur	2
I	Forskningsmetode og teori	5
2	Forskningsstrategi	7
3	Teori	9
3.1	Viktige begreper	9
3.1.1	Universal Multimedia Access	9
3.1.2	Universal Multimedia Experience	10
3.1.3	Quality of Experience	11
3.2	Extensible Markup Language	11
3.2.1	Document Object Model	12
3.3	Metadata	12
3.3.1	Standarder	13
3.4	Hypertext Transmission Protocoll	14
3.5	Online datainnhenting	14
3.5.1	Likert-skala	15

II	Bevisstgjøring og utvikling	17
4	Bevisstgjøring	19
4.1	Utvikling av multimediasystemer	19
4.2	Eksisterende løsninger	21
5	Forslag	23
5.1	Konseptbeskrivelse	23
5.2	Potensielle fordeler og ulemper ved konsept	25
5.3	Kravspesifikasjon	26
5.4	Ønsket brukergrensesnitt	28
6	Utvikling av demo-system	31
6.1	Valg av innhold	31
6.2	Begrensinger i forhold til konsept	32
6.3	Systemmodell	32
6.4	Utviklingsmiljø	33
6.5	Implementasjon av redaktørapplikasjon	37
6.5.1	Import og database	37
6.5.2	Brukergrensesnitt og interaksjon	38
6.6	Implementasjon av sluttbrukerapplikasjon	43
6.6.1	Brukergrensesnitt og interaksjon	43
6.7	Kommunikasjon og tjener	45
III	Evaluering og konklusjon	47
7	Evaluering	49
7.1	Brukertest av demo-systemet	49
7.1.1	Forberedelser	49
7.1.2	Gjennomføring	50
7.1.3	Resultater og analyse	51
7.2	Diskusjon	54
7.2.1	Demo-systemets nytteverdi	54

7.2.2	Validering av kravoppnåelse	55
7.2.3	Kritikk til demo-system	56
8	Konklusjon	59
8.1	Videre utvikling	60
A	Tjener-filer	63
B	Brukertest	65
B.1	Testinstruksjon	65
B.2	Kildekode fra brukertest	66

Figurer

3.1	Systemmodell for “Universal Multimedia Access”	10
3.2	Eksempel på et XML dokument	12
5.1	Systemdiagram av konseptidé	24
5.2	Ønsket brukergrensesnitt redaktørapplikasjon	28
5.3	Skisse av ønsket brukergrensesnitt: Liste og avspilling	29
5.4	Skisse av ønsket brukergrensesnitt: Liste og tilbakemelding	30
6.1	Systemdiagram av demo-system	32
6.2	Eksempel på programmering i Max 6	34
6.3	Eksempel på Android-programmering i Eclipse	35
6.4	Eksempel på oppsett av server med Xampp	36
6.5	Import av innhold til redaktørapplikasjonen	37
6.6	Utdrag av taggestruktur	38
6.7	Hovedvindu i redaktørapplikasjon	39
6.8	Oppsett av tilbakemeldingsvindu	41
6.9	Vindu for visning av sluttbrukerlogg	42
6.10	Brukergrensesnitt sluttbrukerapplikasjon: presentasjonsvindu	44
6.11	Brukergrensesnitt sluttbrukerapplikasjon: tilbakemeldingsvindu	45
6.12	Modell for kommunikasjon via tjener	45
7.1	Brukertest gjennomføring	51
7.2	Resultat brukertest	52

Kapittel 1

Introduksjon

Mennesket har siden tidenes morgen vært aktive historiefortellere. I moderne tider med datamaskiner og internett, har historiene blitt mer og mer digitalisert. De fortelles gjerne i form av en kombinasjon av multimedialt innhold som tekst, bilde, lyd og video som har ført til en enorm vekst i multimedialindustrien. I senere år skyldes i stor grad dette den store økningen i teknologier som fasiliter trådløs kommunikasjon, samt tilgjengelighet i form av mobile nettverk med stadig økende båndbredde. I Cisco sin seneste prognose[1] anslåes det at innen 2018 så vil to tredjedeler av all trafikk på mobile nettverk være video. Allerede i 2012 overskred video-trafikken 50%. Private og mindre profesjonelle aktører har i dag relativt gode kameraer og lydopptakere på sine smarte enheter, og utgjør sammen med større aktører noe av årsaken til en stor økning i tilførselen av innhold på nett. Med dette store biblioteket av innhold oppstår viktigheten av å tilføre beskrivende informasjon, kalt metadata, til innholdet for å øke tilgjengelighet for brukeren. Brukerne har gjerne preferanser til hva de ønsker å se eller lytte til, og metadata er dermed en viktig forutsetning for tilgjengelighet. Etterhvert har metadata om multimedialt innhold sammen med utviklingen av trådløs teknologi, åpnet for mye forskning på hvordan metadata og interaksjon mellom enheter kan brukes på nye måter for å skape bedre brukeropplevel-

ser, og nye bruksområder. Det eksisterer i dag forskjellige løsninger som “Recommender Systems” [23], “Session Mobility” [20] og “Second Screen” [22] som er interessante tilnærminger til dette temaet.

1.1 Mål for oppgaven

Det finnes i dag flere gode tjenester for konsumering av innhold, der innholdet tradisjonelt distribueres direkte fra en tilbyder til en sluttbruker. Denne oppgaven ønsker å utforske hvordan informasjon som beskriver multimedialt innhold, kan brukes som en ressurs for å utvide funksjonaliteten til dagens systemer for innholdsdistribusjon over nettverk. Dette skal gjøres gjennom å introdusere et intelligent ledd i form av en applikasjon mellom innholdstilbyder og sluttbruker. Denne applikasjonen skal ha en tilbakekobling fra sluttbrukeren. Til sammen skal det utvikles et multiplattform system som demonstrerer denne idèen. Målet for oppgaven er videre delt opp i 3 delmål:

Delmål 1: Designe og implementere et multiplattform demo-system som innfører et intelligent ledd i en tradisjonell multimedia-distribusjonsmodell.

Delmål 2: Gjennomføre en brukertest for videre evaluering av om systemet fungerer som forventet, og se på systemets nytteverdi.

Delmål 3: Validere demo-systemet opp mot kravspesifikasjon og forventninger.

1.2 Oppgavens struktur

Opgaven er delt opp i tre hoveddeler. I første del presenteres forskningsmetode og relevant teori for oppgaven. Den andre delen inneholder en beskrivelse av hvordan oppgaven har blitt løst, og hvilke resultater forskningsarbeidet

har produsert. Kapittel 4 inneholder en bevisstgjøring av problemet oppgaven søker å løse, etterfulgt av kapittel 5 som inneholder et forslag til en løsning i form av et konsept for et demo-system. Videre vil kapittel 6 beskrive hvordan dette løsningsforslaget ble realisert, altså hvordan konseptet ble implementert. I Del III vil kapittel 7 gå gjennom en brukertest for å kunne evaluere demo-systemet, og til slutt avrunde evalueringen av systemet med diskusjon rundt resultatene av forskningsarbeidet. Kapittel 8 inneholder en konklusjon og tanker om videre arbeid.

Del I

Forskningsmetode og teori

Kapittel 2

Forskningsstrategi

Gjennom arbeidet med problemstillingen i denne oppgaven, er det tatt utgangspunkt i en fremgangsmåte som Brinary J Oates skriver om i sin bok “Researching Information Systems and Computing” [2]. I kapittelet “Design and Creation” beskriver Oates en modell for utførelse av design- og utviklingsforskning. Denne prosessen deles opp i fem steg, og heter “Learning via Making”. De fem stegene er: Bevisstgjøring(Awareness), Forslag(Suggestion), Utvikling(Development), Evaluering(Evaluation) og Konklusjon(Conclusion).

Bevisstgjøring: Her skal problemet anerkjennes og forstås. Dette kan blant annet gjøres gjennom å se på relatert litteratur der man ser hva som uttrykkes av forslag til videre utvikling, eller gjennom forbedringsønsker fra utøvere av teknologi.

Forslag: Denne delen dreier seg om den kreative prosessen som medhører for å utforme en tentativ ide til hvordan problemet skal adresseres.

Utvikling: Under utvikling skal den tentative ideen implementeres. Hvordan dette gjøres vil avhenge noe av IT-løsningen som skal utvikles.

Evaluering: Etter at utviklingssteget er gjennomført skal løsningen undersøkes for hvilken nytteverdi, og avvik fra forventning den har.

Konklusjon: Her blir resultatene av designprosessen samlet og den tilegnede kunnskapen blir identifisert.

Utviklingen følger en steg-for-steg prosess, men den trenger ikke å følges slavisk. Flere steder i denne prosessen kan man oppdage at det som er gjort tidligere kan utbedres. Ved å tenke ut tentative løsninger og idèer kan det oppstå en større bevisstgjøring av problemstillingens natur. Utvikling av en designidé kan lede til økt forståelse av problemet og man kan oppdage nye mulige løsninger. Dermed blir prosessen mer løs og iterativ, og man lærer gjennom hele utformingsprosessen.

Kapittel 3

Teori

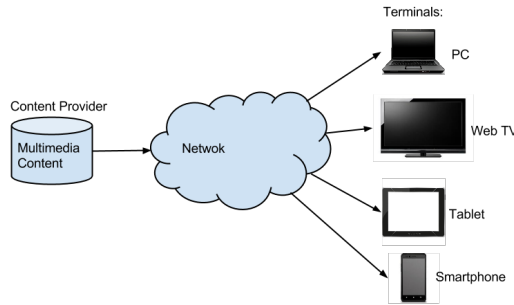
3.1 Viktige begreper

I denne delen presenteres noen viktige begreper innenfor multimedia som danner et godt fundament for forståelsen av oppgavens fagfelt.

3.1.1 Universal Multimedia Access

“Universal Multimedia Access” [3], forkortet UMA, referer til evnen til å sømløst kunne få tilgang til et vilkårlig multimediaminnhold hvor som helst, når som helst og på hvilken som helst plattform eller enhet. Dette kan kreve en form for adaptasjon av innholdet, der det må transkodes til et passende format eller bitrate, eller i form av endring av modalitet fra for eksempel tale til tekst. Fokuset for UMA er i stor grad å endre innholdet for å møte begrensninger til en brukers terminal eller nettverk.

Figur 3.1 viser hvordan et UMA-system fungerer. Innholdet oppbevares først i sin beste kvalitet hos en innholdsforsørger. Fra forsørgeren distribueres



Figur 3.1: Systemmodell for “Universal Multimedia Access”

innholdet over et nettverk til terminaler. Kvaliteter som bitrate, format og modalitet endres etter nettverkets og terminalens begrensninger.

3.1.2 Universal Multimedia Experience

“Universal Multimedia Experience” [3], forkortet UME, referer til at en bruker burde ha en likeverdig, informativ opplevelse når som helst, og hvor som helst. En slik opplevelse vil typisk bestå av flere former for multimedia innhold. Innholdet vil gjennomgå en adaptasjon på samme måte som for UMA, men fokuset ligger i at adaptasjonen styres av grensene som forsikrer at brukeren har en verdifull, informativ opplevelse. Brukeren er dermed i sentrum, og terminal og nettverk blir mer et hjelpende redskap i motsetning til UMA der dette er styrende.

3.1.3 Quality of Experience

“Quality of Experience”[4] eller QoE er en betegnelse på et subjektivt mål av hvordan en sluttbruker oppfatter kvaliteten av en applikasjon eller levert tjeneste. Dette inkluderer alt fra tekniske kvaliteter til menneskelig oppfatning. Dermed er det mange faktorer som spiller inn, og standarder for å best mulig definere mål på hva som er god QoE kan være utfordrende. Hos “International Telecommunication Union”¹, som er en viktig organisasjon for global standardisering innenfor telekommunikasjon, har dette blitt forsket mye på. En anerkjent standard som ITU-T BT.500[5] er et eksempel på en teststandard for subjektiv måling av QoE.

3.2 Extensible Markup Language

XML [13] er en forkortelse for “Extensible Markup Language” som er et enkelt og fleksibelt tekstformat for transport og lagring av data. Det er utviklet av “World Wide Web Consortium”², og er en delmengde av SGML (“Standard Generalized Markup Language”). XML skulle originalt møte utfordringer knyttet til stor-skala elektronisk utgivelse, men brukes i dag til utveksling av en rekke forskjellige data på nett og andre steder. Grunnet enkelheten rundt å strukturere data og beskrivelser av data i det som kalles elementer, er XML populært i multimediasammenheng. XML stiller strenge regler til syntaks, noe som er viktig for at programvare skal kunne lese eller analysere informasjonen inneholdt i taggete elementer. Dokumenter som følger disse reglene kalles “well formed documents”.

Et eksempel på koding i XML er vist i figur 3.2. Dokumentet starter med en XML deklarasjon som spesifiserer versjon og encoding som brukes. Videre er dokumentet som nevnt over bygget opp av elementer som blir definert av en start- og slutt-tag. Informasjonen til elementet er det som ligger innenfor disse taggene. Et element kan inneholde forskjellige typer informasjon

¹<http://www.itu.int/en/about/Pages/default.aspx>

²<http://www.w3.org/>

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to> Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Figur 3.2: Eksempel på et XML dokument

som tekst, attributter og andre elementer. Når elementer inneholder andre elementer dannes en hierarkisk struktur.

3.2.1 Document Object Model

En DOM (“Document Object Model”) definerer en standard måte for å akseptere og manipulere data ³. Det er en multiplattform og språkuavhengig konvensjon for HTML, XHTML og XML dokumenter. En DOM for XML representerer dokumentet som en trestruktur, og elementene aksesserer i et DOM tre. For at en programmerer skal kunne gjennomføre disse manipulasjonene brukes et definert API (“Application Programmable Interface”).

3.3 Metadata

Generelt kan metadata beskrives som data om data. Det forsyner informasjon om et objekt sitt innhold. Gilliland[9] beskriver metadata som

the sum total of what we can say about any information object at any level of aggregation, in a machine understandable representation.

Der et informasjonsobjekt er

³http://www.w3schools.com/xml/xml_dom.asp

anything that can be addressed and manipulated by a human, or a system as a discrete entity.

Metadata kan klassifiseres etter funksjonen det er ment å støtte (deskriptiv, strukturell, administrativ, rettighetshåndtering), eller sitt nivå av semantisk abstraksjon (lavnivå vs. høynivå)[11]. Lavnivå er typisk automatisk generert teknisk metadata som har mindre informativ verdi for sluttbrukere. Høynivå semantisk rik metadata har mer informativ verdi og beskriver semantiske enheter i en fortellende verden som objekter, hendelser, konsepter, semantiske tilstander, semantiske steder, semantiske tider, sammen med deres attributter og relasjoner. Denne høynivå-delen er ofte brukergenerert. For metadata som beskriver multimedialt innhold går et viktig skille mellom metadata som beskriver kontinuerlig (audio, video) og ikke-kontinuerlig (tekst, bilde) innhold. I kontinuerlig innhold har man en tidslinje der metadataen kan være temporalt knyttet opp til et punkt eller en sekvens[10].

3.3.1 Standarder

Det finnes mange måter å lagre metadata på. For å prøve å unngå at alt lagres på forskjellige måter er det opprettet standarder som fastsetter hvordan forskjellig typer informasjon skal representeres. En standard i metadata-sammenheng er et rammeverk for å beskrive mening og semantikk til data. Bruk av standarder tillater enkel utveksling av data mellom forskjellige aktører. Typisk defineres et skjema som brukeren av standarden må følge[12]. En utbredt standard for multimedialt innhold er *EBU Core* [6]. Denne standarden baserer seg på Dublin Core-standard[8], som originalt ble laget for at forfattere skulle beskrive sine web-ressurser. EBU Core ble først publisert i år 2000 som et sett definisjoner for audioarkiver som et tillegg til Dublin Core[7]. EBU Core er i dag en viktig metadata-standard for multimedia og tilbyr en stor bredde av muligheter for beskrivelse av innhold.

3.4 Hypertext Transmission Protocoll

Http(“Hypertext Transmission Protocol”)[14] er en applikasjonsnivåprotokoll for distribuerte-, samarbeids- og hypermedia-informasjonsystemer. Denne protokollen er i dag den mest brukte for utveksling av informasjon over internett, og fungerer som en forespørsel/respons protokoll i et klient/tjener miljø. En klient sender en forespørselmelding til en tjener i form av en forespørselsmetode, URI og protokollversjon etterfulgt av meldingen. Tjeneren behandler denne forespørselen og sender en responsmelding til klienten med fullføringstatus, og eventuelt etterspurt innhold.

Blant de viktigste forespørselsmetodene er “GET” og “POST”. Med “GET”-forespørselen etterspør man en representasjon fra en spesifisert kilde. Man vil altså kunne motta data med denne metoden. En “POST”-forespørsel gir ordre til tjeneren om å motta instansen innkapslet i forespørselen som en ny underordnet av ressursen identifisert ved forespørsels-URI’en. Et bruksområde kan dermed være å tilføre innhold til en allerede lagret ressurs hos en tjener.

3.5 Online datainnhenting

Med en verdensbefolkning som er mer og mer online har det blitt en populær forskningsmetologi å samle inn data via mail eller web-baserte spørreundersøkelser[15]. Sammenlignet med tradisjonelle “papir og blyant”-undersøkelser er det flere store fordeler med denne formen for datainnhenting som ressursbesparelser, redusert responstid, enklere for bruker å besvare, større fleksibilitet med tanke på format, og et potensielt stort tillegg i antall besvarer grunnet utvidet utstrekning. Online datainnhenting har mange store positive sider ved seg, men det er også flere bekymringer knyttet til denne formen for datainnhenting. Uvisshet om et representativt utvalg deltar, dårlig svarprosent og målingsfeil knyttet til psykometri i overgangen fra tradisjonelle spørreundersøkelsesmetoder er ikke kjent. Svarprosent har vist

seg å bli et mindre problem ettersom evnen til å betjene datamaskiner øker i befolkningen. De nevnte faktorene leder alle til usikkerhet rundt hvorvidt dataens integritet er opprettholdt ved bruk av online datainnhenting, og må tas inn i vurderingen før det benyttes.

3.5.1 Likert-skala

Det er vanlig ved innsamling av data i undersøkelser, å definere en psyko-metrisk skala for å uttrykke graden av enighet ved en uttalelse eller påstand. Likert-skalaen[16] er nettopp en slik skala, og settes vanligvis opp som 5 punkter som beveger seg fra “svært enig” på en side til “svært uenig” på den andre, med “hverken enig eller uenig” i midten. Denne skalaen er i dag anerkjent, og ble første gang publisert i 1932 av sosiolog Dr. Rensis Likert.

Del II

Bevisstgjøring og utvikling

Kapittel 4

Bevisstgjøring

I forskningsmetoden beskrevet i del I er første steget for utvikling av en IT-løsning bevisstgjøring. Derfor blir det i denne delen tatt for seg dagens situasjon for utviklingen av UMA, relevant teknologi, og eksisterende løsninger innenfor oppgavens faglige omfang.

4.1 Utvikling av multimediasystemer

Allestedsnærværende multimedia har i dag nærmest blitt en realitet takket være en enorm utvikling de siste to tiårene. Utviklingen av internett, og etter hvert mobile nettverk, med tanke på kapasitet og utstrekning, har vært veldig viktige faktorer for å muliggjøre UMA med tilgang hvor som helst. Opprettelsen av standarder for lyd (som MPEG-2, AAC), video (som H.264, H.265) og protokoller for strømming av disse, har også vært viktige bidrag.

Etter hvert som mange utfordringer knyttet til UMA har blitt løst, har man fått en økt forståelse for at brukerens opplevelse av et innhold avgjøres av mer enn nettverk og innholdsadapsjon. Neste steg ble dermed

UME-konseptet som fyller dette tomrommet. UME inkluderer også ikke-tekniske faktorer som eksempelvis psykologien knyttet til en brukers opplevelse av multimedia-informasjon. Denne endringen av fokus sammenlignet med UMA flytter endepunktet i systemet fra terminalen, og over til brukeren som konsumerer innholdet.

For å kunne si noe om hvor vellykket et UME-system er, må man på en eller annen måte kunne måle hvordan oppfatningen av en brukeropplevelse har vært. Dette har ledet til uttrykket QoE (“Quality of Experience”) som prøver å oppnå akkurat dette. QoE har de siste årene vært et populært forskningsområde der man blant annet har laget evalueringsmodeller for QoE[17][18]. Det forsøkes også å lage objektive mål som reflekterer QoE, noe som for eksempel muliggjør å lage autonome systemer for nettverkshåndtering basert på QoE[19].

For å bedre mulighetene for sømløs strømming av innhold uavhengig av posisjon er “Session Mobility”[20] en teknologi som har oppstått. Med “Session Mobility” forsøker man å oppnå at en brukers avspilling av innhold skal være allestedsnærværende uavhengig av om ytre forhold endrer seg. Dette kan være fysisk mobilitet i form av nettverksendringer, samt service mobilitet i form av at man kan fortsette avspilling på alle enheter tilgjengelig for brukeren. En teknologier som muliggjør dette er SIP (“Session Initial Protocol”)[21]. Dette er en protokoll for overføring av initielle data ved bytte av dings, slik at avspilling kan fortsette. Også introduksjon av sensorteknologi er testet for å skape en sømløs “Session mobility”¹. Et annet felt det forskes mye på er såkalt “Second Screen”-teknologi[22] som utnytter at privatpersoner eier stadig flere enheter. Denne teknologien sørger for interaksjon med innholdet presentert på en TV gjennom bruken av en sekundær håndholdt skjerm. Disse ovennevnte formene for kommunikasjon mellom enheter kan minne litt om det som implementeres i denne oppgaven, men de tjener en annen funksjon enn oppgavens mål.

¹<http://www.nist.gov/smartspace/>

4.2 Eksisterende løsninger

Det finnes i dag flere fagfelt med løsninger som tar tak i viktige utfordringer knyttet til metadatautnyttelse, og datainnhenting.

En mye utbredt metode for best mulig tilpasning av brukeropplevelser er bruk av såkalte “Recommender Systems”[23]. Disse samler sammen metadata som geografisk posisjon og brukeraktivitet. Til slutt gjennomføres en automatisert analyse som leder til at bruker blir eksponert for mest mulig relevant og verdifull informasjon. Arbeid med å forbedre disse systemene er et stort forskningsområde, og en del av forbedringene ligger i evnen til å benytte seg av bredden og semantikken i metadata på en effektiv måte. Dette kan være informasjon som posisjon[24], samt utnyttelse av høyere lag av semantisk informasjon. Bruken av “Recommender Systems” på multimedialt innhold er populært og brukes i store “Over-the-Top”-tjenester som *YouTube*² og *Netflix*³.

For at man skal kunne utnytte metadata på en god måte kan det være en fordel å ha generert noen gode data i utgangspunktet. Innsamling av data fra brukere gjennom web-baserte tjenester er et populært felt, med flere etablerte spørreundersøkelsestjenester[25] som *SurveyMonkey*⁴, og *Google Consumer Survey*⁵. Gjennom disse verktøyene kan man potesnielt nå ut til en veldig stor brukermasse, samt at de tilbyr gode automatiserte tjenester for fremstilling og analyse av resultater. Verktøyene har et ganske generelt bruksområde, og flere tjenester begynner nå å kunne inkludere multimedialt innhold til spørsmålene i undersøkelsen. Det har gjerne en begrenset formatstøtte, men muligheten er der. En ulempe med denne formen for datainnhenting er at brukeren som besvarer spørreundersøkelsen ikke har noen særlig frihet i form av at man ofte må besvare alle spørsmål før undersøkelsen er over.

²<https://www.youtube.com/yt/about/>

³<http://ir.netflix.com/>

⁴https://no.surveymonkey.com/?ut_source=header

⁵<https://www.google.com/insights/consumersurveys/home>

Kapittel 5

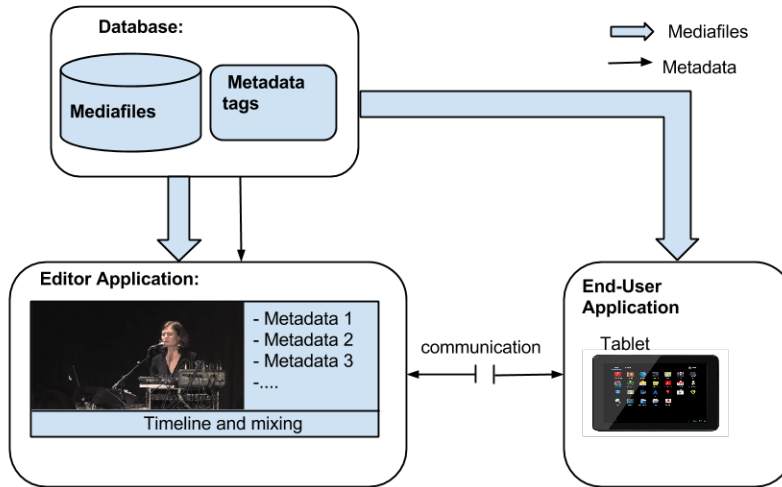
Forslag

Dette kapittelet tar for seg forslaget til et system som kan knyttes opp til en utvidet bruk av et UMA-system. Innledningsvis presenteres et konsept som videre utvikles til en designidé for et demo-system som blir utgangspunktet for implementasjonen.

5.1 Konseptbeskrivelse

Konseptidéen er et resultat av et ønske fra Andreas Bergsland som er Postdoktor ved Institutt for musikkteknologi på NTNU. Litt i tråd med UME tankegangen ønskes det i denne oppgaven å utarbeide et system som skaper gode brukeropplevelser med semantisk metadata knyttet til multimedialt innhold som hjelp. Det ønskes også å få dette systemet til å videre berike det multimediale innholdet.

Figur 5.1 viser en modell av konseptidéen. Hvis man ser denne figuren sammenlignet med figur 3.1, oppdager man at dette systemet skiller seg ut fra standard UMA ved at en ny applikasjon er lagt til i leddet mellom databasen og sluttbruker(terminal). En database med innhold distribuerer dette



Figur 5.1: Systemdiagram av konseptidé

trådløst til to typer applikasjoner, der den ene applikasjonen har full tilgang til både multimedialt innhold og metadata, mens den andre kun har tilgang til det multimediale innholdet. Disse leddene er valgt å døpes henholdsvis redaktørapplikasjonen og sluttbrukerapplikasjonen. Det har vært viktig at disse applikasjonene skal være multiplattform, så det er tenkt at redaktørapplikasjonen skal være en applikasjon for PC, og sluttbrukerapplikasjonen skal kunne kjøres på et nettbrett. Redaktørapplikasjonen har som funksjon å presentere innhold og metadata på en god måte, samt å styre hva sluttbrukerapplikasjonen skal ha tilgang til av metadata. Dette vil være med på å styre utformingen av sluttbrukerapplikasjonen. Applikasjonen skal også kunne samle inn informasjon om aktiviteten til sluttbrukeren. Metadataen fra databasen (se figur 5.1) skal i første omgang være informasjon som er annotert på forhånd, og som har en semantisk betydning for innholdet. Metadataen er dermed potensielt berikende for det multimediale innholdet, og tanken er at sluttbrukeren skal bidra til videre berikelse av

innholdet gjennom å automatisk sende ulike tilbakemeldinger og aktivitetsloggføring tilbake til redaktøren. Fokuset for dette systemet ligger dermed i konfigurasjonsmulighetene mellom sluttbrukerapplikasjonen og redaktørapplikasjonen. For å muliggjøre dette må det opprettes en form for toveis kommunikasjon mellom plattformene hver applikasjon kjøres på. Denne er tenkt å være trådløs i form av en enkel web-tjener.

Videre følger en definering av rollene til brukeren av hver applikasjon. Det er tenkt at redaktøren er en innholdsformidler med god kjennskap til innholdet og kan potensielt sett ha mange agendaer. I denne oppgaven er redaktøren definert som en person med et forskningsmotiv. Denne forskeren har behov for et verktøy for representasjon av innholdet basert på metadata. Redaktøren har også et forskermotiv for datainnsamling fra en brukergruppe. Grunnet hensyn som f.eks. anonymisering ønsker ikke redaktøren å dele alle metadata med sluttbrukeren og trenger dermed en mulighet til å styre de metadata som leveres til sluttbruker. Sluttbrukeren er en person som på en eller annen måte har en interesse av både å konsumere og vurdere innholdet i sluttbrukerapplikasjonen.

5.2 Potensielle fordeler og ulemper ved konsept

Basert på den tidligere bevisstgjøringen og tanker rundt konseptet kan man tenke seg til flere positive sider ved dette konseptet. Siden det i dag er flere private og mindre profesjonelle aktører som produserer innhold, kan det være nyttig for disse å kunne distribuere dette i et slikt system. Sammenlignet med mer tradisjonelle former for datainnhenting innenfor forskning er det her tenkt at overgangen til dette systemet kan lede til tids- og ressursbesparelser. Ved datainnhenting til redaktør fra en smart enhet åpner man for muligheten til å gi passive tilbakemeldinger som ved risting av enheten eller lignende.

Ved innhenting av data vil man som forsker ofte ha kontrollerte omgivelser. For dette konseptet er omgivelsene vanskelig å styre, og det kan også

være utfordrende å kontrollere hva slags type mennesker som gir tilbakemeldinger i sluttbrukerapplikasjonen. Det positive er ganske klart at man potensielt sett når ut til en stor gruppe mennesker hvis denne linken mellom redaktør og sluttbruker foregår trådløst over et nettverk. På denne måten vil forhåpentligvis mengden av data veie opp for usikkerhetene rundt ukontrollerte omgivelser ved datainnhenting. Andre forhold som at brukerne av applikasjonene trenger tilgang til trådløst nettverk, og mest sannsynlig en ganske god båndbredde, bidrar til en potensiell begrensning av brukermassen. Det er også en ulempe hvis metadataen er annotert på et språk som sluttbrukeren ikke forstår.

5.3 Kravspesifikasjon

For å kunne sette noen klare mål for hva man vil utrette med systemet ble det utformet en kravspesifikasjon. Ut fra konseptet er det tenkt at det skal utvikles to applikasjoner. En applikasjon til en PC, og en applikasjon til et nettbrett. Det er vanlig å definere krav etter funksjonelle krav som beskriver hvordan en applikasjon skal oppføre seg, samt ikke-funksjonelle krav som sier mer om hvordan en applikasjon skal være. Kravene varierer i fra å være mer generelle til temmelig spesifikke.

Redaktørapplikasjon

Funksjonelle krav:

1. Automatisk import av multimediafiler med tilhørende metadata til redaktørapplikasjonen.
2. Få metadata visualisert i brukergrensesnittet som også skal kunne benyttes mot sluttbrukerapplikasjonen.
3. Tilordne metadata-annoterte temporale utdrag av innholdet til en liste som sluttbrukerapplikasjonen trådløst skal ha tilgang til.

4. Velge hva brukeren skal ha mulighet til å gi av tilbakemelding vedrørende innholdet.
5. Vise en loggføring av aktivitet og tilbakemeldinger fra sluttbrukerapplikasjonen.
6. Mulighet for å navigere seg rundt i en hierarkisk metadatastruktur med metadata-annoterte temporale utdrag.

Ikke-funksjonelle krav:

1. Applikasjonen skal være så intuitiv som mulig, men det forventes at bruker må få en liten opplæring.
2. Applikasjonen skal kunne navigeres rundt i sømløst, og bruker skal oppleve rask respons ved interaksjon.

Sluttbrukerapplikasjon

Funksjonelle krav:

1. Bruker skal få tilgang til en liste med audiovisuelt innhold.
2. Bruker skal kunne spille av innholdet etter eget ønske.
3. Bruker skal kunne gi tilbakemelding på hvert element i innholdslisten
4. Tilbakemelding skal kunne gis både passivt i form av f.eks. risting av nettbrett, og aktivt i form av f.eks. en kommentar som bruker selv skriver.
5. Muligheter for å spille av innhold i en fullskjermsmodus.

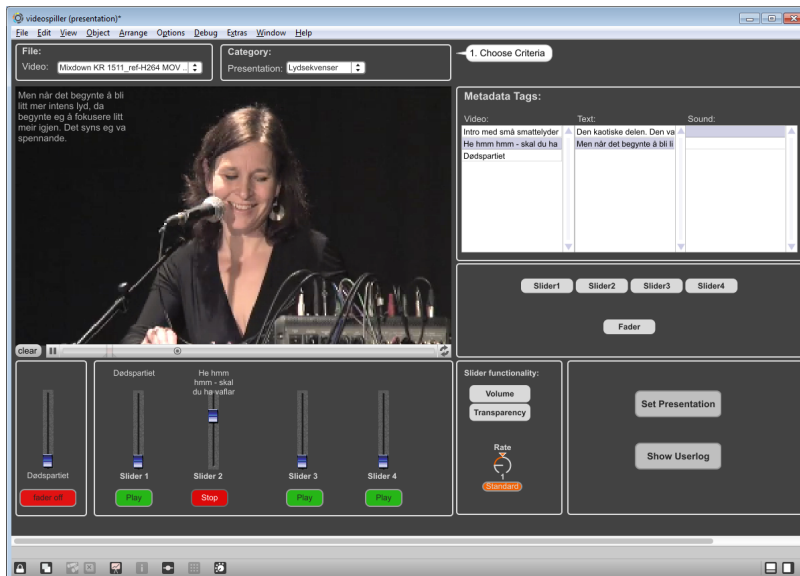
Ikke-funksjonelle krav:

1. Applikasjonen skal fremstå minimalistisk og intuitiv slik at den kan brukes uten opplæring.

- Applikasjonen skal kunne navigeres rundt i sømløst, og bruker skal oppleve rask respons ved interaksjon.

5.4 Ønsket brukergrensesnitt

I denne delen vises noen figurer med skisser av brukergrensesnitt som er utarbeidet før implementasjonen. Disse skissene har dannet utgangspunktet for utformingen av de implementerte brukergrensesnittene i applikasjonene.

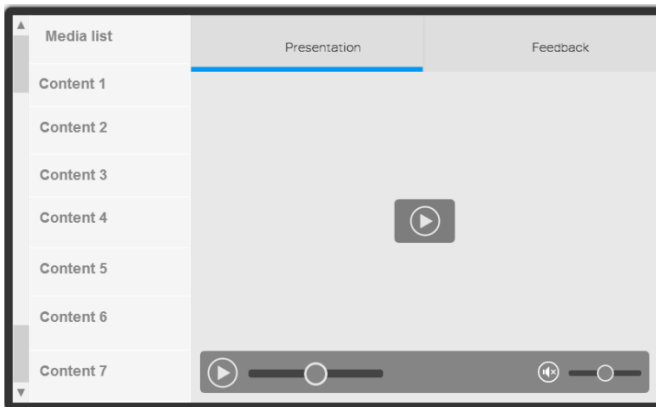


Figur 5.2: Ønsket brukergrensesnitt redaktørapplikasjon

Figur 5.2 viser grovt hvordan det er ønsket at redaktørapplikasjonen skal se ut. For skisseringen er det tatt utgangspunkt i en applikasjon som er blitt laget av utvikleren tidligere, der ny funksjonalitet som knapper enkelt er lagt til i et visuelt programmeringsprogram. Den nye funksjonaliteten

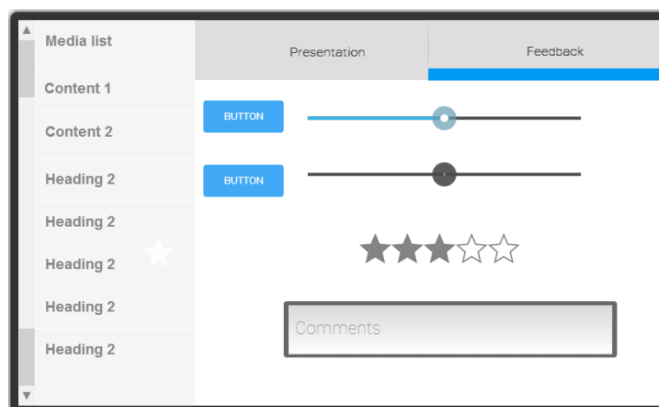
dreier seg om at man ved hjelp av knapper skal kunne åpne vinduer som gir mulighet til å konfigurere tilbakemeldingsvinduet i sluttbrukerapplikasjonen, samt en knapp for å vise et vindu med brukerlogg av aktiviteten i sluttbrukerapplikasjonen. Disse knappene er henholdsvis “Set Presentation” og “Show Userlog” (nederst til høyre i figur 5.2). Over dette er en visning av metadatainnholdet til en valgt fil.

Figur 5.3 og 5.4 viser en grov skisse for hvordan det er ønsket at sluttbrukerapplikasjonen skal se ut. Siden et av målene for applikasjonen er at brukergrensesnittet skal være intuitivt og enkelt å forstå, er det valgt å ha et enkelt utseende med få knapper, og kun to forskjellige visninger. En visning for å presentere innholdet, og en visning for å gi tilbakemelding på innholdet.



Figur 5.3: Skisse av ønsket brukergrensesnitt: Liste og avspilling

Figur 5.3 viser en presentasjonsvisning der bruker kan se en innholdsliste og en videovisning. Figur 5.4 viser en tilbakemeldingsvisning der bruker kan se innholdsliste og et tilbakemeldingsvindu som skal bidra til metadataberikelse av innholdet. Ved hjelp av fanen over både videovisning og tilbakemeldingsvindu får man byttet mellom disse to visningene.



Figur 5.4: Skisse av ønsket brukergrensesnitt: Liste og tilbakemelding

Kapittel 6

Utvikling av demo-system

Denne delen tar for seg utvikling og implementering av demo-systemet. Under utviklingen har hovedfokuset vært å implementere slik at kravspesifikasjonen er overholdt, da med spesielt fokus på funksjonelle krav.

6.1 Valg av innhold

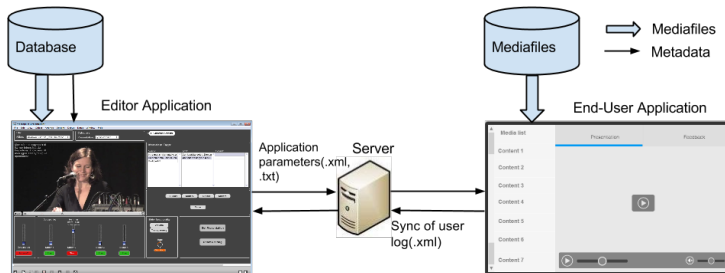
For å skape et best mulig sluttprodukt, er dette systemet avhengig av en rikt annotert database for å være interessant. Databasen som er utgangspunktet for implementasjonen er en multimedia-database fra et av Andreas Bergsland sine prosjekter. Innholdet er video og lyd fra en musikkfremføring med friimprovisert ensemblemusikk. Av videoinnhold er fremføringen filmet fra to vinkler, samt at lydkanalene fra hvert instrument eksisterer som egne lydfiler. Av metadata er innholdet manuelt annotert etter EBU Core standarden. Informasjonen i hver metadatafil innebefatter primært en beskrivelse av filen, og tagger som er temporalt knyttet til områder i filenes tidslinje. Disse temporalt annoterte områdene blir videre i denne oppgaven kalt utdrag. Utdragene er kategorisert slik at informasjonen danner en hierarkisk struktur.

6.2 Begrensinger i forhold til konsept

Grunnet tidsbegrensning og oppgavens omfang ble ikke alle sider av konseptet mulig å implementere. Siden strømming av multimedialt innhold ikke er et av fokusområdene for denne oppgaven ble innholdet derfor gjort tilgjengelig lokalt på enhetene som har blitt brukt. Det har også blitt nedprioritert å se på muligheter for passive tilbakemeldinger i sluttbrukerapplikasjonen som var en del av funksjonelt krav nummer 4. Også krav nummer 5 om muligheter for fullskjermmodus har ikke blitt ferdig implementert.

6.3 Systemmodell

Før implementasjonen går gjennom mer i detalj, forklares først demo-systemets overordnede dataflyt. Figur 6.1 viser en modell av systemet som er implementert.



Figur 6.1: Systemdiagram av demo-system

Databasen som består av multimedia- og metadatafiler ligger lagret lokalt på redaktøren sin enhet, og importeres automatisk til redaktørapplikasjonen. Ut ifra valg redaktøren foretar seg i applikasjonen sendes konfigurasjonsparametere til sluttbrukerapplikasjonen via en tjener som kjøres på redaktørens maskin. Ved oppstart av sluttbrukerapplikasjonen vil applikasjonen

automatisk hente inn informasjon fra disse filene for styring av brukergrensesnittet. Primært dreier dette seg om informasjon rundt hvilke utdrag som skal inkluderes, samt hva slags type tilbakemeldinger som skal kunne gis. I sluttbrukerapplikasjonen blir aktivitet og tilbakemeldinger lastet opp til tjeneren som en brukerloggfil. Informasjonen fra denne filen hentes inn ved at redaktøren velger å få vist en brukerlogg. Siden det ikke foregår noen strømming av multimedialt innhold, ligger innholdet også lagret lokalt på enheten til sluttbrukeren.

6.4 Utviklingsmilø

I denne delen introduseres og begrunnes valget av programvaren som er brukt i utviklingen av demo-systemet. Enhetene applikasjonen skal implementeres på er en PC med “Windows 7” som operativsystem, og et nettbrett med “Android 4.3 Jelly Bean” som operativsystem. Grunnet en relativt omfattende implementeringsoppgave har det under valget av programvare blitt vektlagt å bruke utviklerprogrammer som tillater hurtig implementering, og som enkelt lar funksjonell programmering realiseres til et brukergrensesnitt. Kriterier som at programmene er godt dokumentert og gratis har også vært viktig. Den valgte programvaren for verktøyutviklingen ble:

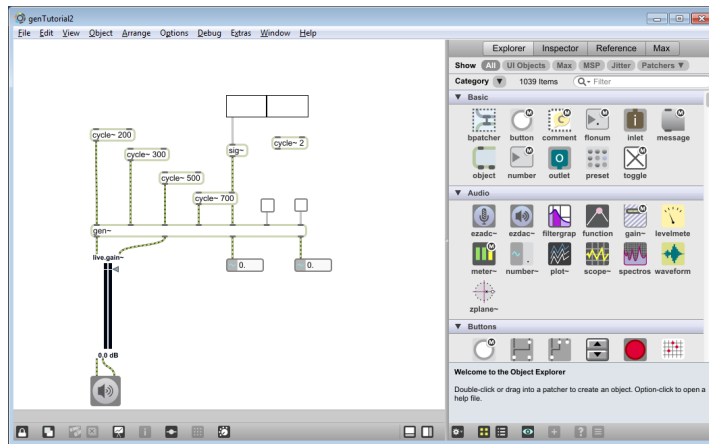
Programvare

- Max 6
- Android SDK Bundle, Eclipse
- Xampp

*Max 6*¹ baserer seg på visuell programmering og er laget for behandling av primært lyd, men i senere år er det også lagt til programpakken *Jitter* som tar seg av video/matrise-behandling. “Objects” og “Messages” er programmets to hovedkomponenter, og programmeringen foregår gjennom

¹<http://c3ycling74.com/products/max/>

å grafisk koble disse sammen. Programmet er veldig godt dokumentert, og det er stor aktivitet på brukerforum. Denne visuelle programvaren er også et naturlig valg da det vil være enklere og mer tidsbesparende å implementere applikasjons-idèer i forhold til et mer lavnivå programmeringsspråk som JAVA eller C. Et eksempel på hvordan programmering i Max 6 ser ut er vist figur 6.2.



Figur 6.2: Eksempel på programmering i Max 6

Det skal nevnes at Max ikke er åpen kildekode, men har en åpen kildekode-konkurrent ved navn *Pure Data*². Det at Max er et kommersielt produkt gjør at flere hensyn er tatt til enkel og intuitiv programmering. Dette argumentet, sammen med større sikkerhet i forhold til fremtidig vedlikehold gjorde Max til et mer attraktivt valg enn Pure Data. Apple sin *QuickTime player*³ er nødvendig da Max 6 bruker deler av dette programmet til videoavspilling.

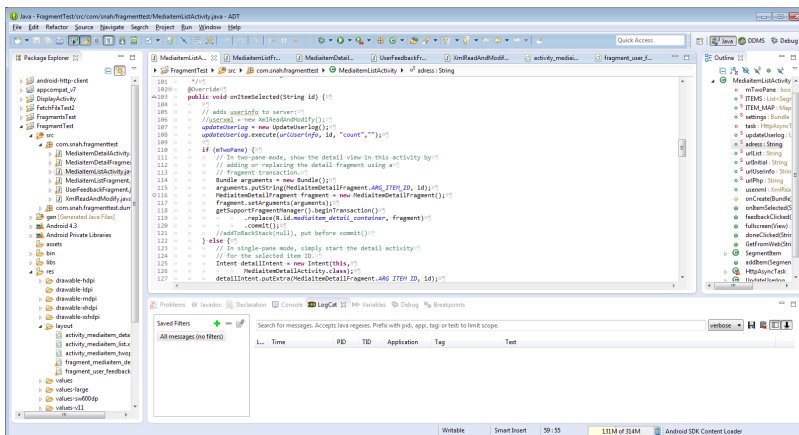
*Android SDK*⁴ består av en rekke verktøy og API-biblioteker for Android-

²<http://puredata.info/>

³<http://www.apple.com/quicktime/what-is/>

⁴<http://developer.android.com/sdk/index.html>

utvikling på PC. For selve programmeringen brukes *Eclipse*⁵ som er en åpen utviklingsprogramvare der Android har inkludert sine biblioteker for Android utvikling. Android baserer seg på programmeringsspråket JAVA⁶ der de har laget et rammeverk og egne biblioteker som er importert i Eclipse. Med JAVA får man i hovedsak implementert funksjonaliteten i programmet. For å sette et brukergrensesnitt med objekter som listevisninger, knapper osv. opprettes dette primært ved hjelp av en egen XML-fil for hvert visningsvindu. Hvordan det kan se ut å programmere i Eclipse er vist i figur 6.3. For å lage en hel applikasjon med brukergrensesnitt må man opprette et prosjekt i Eclipse. Dette prosjektet er satt sammen av en hel del filer og mapper, noe marginen i venstre del av figur 6.3 indikerer. De viktigste filene for utvikling er JAVA-filene, XML-filene som definerer layout og en manifest-fil for prosjektet. Android SDK ble valgt som utviklervertøy grunnet at det er godt dokumentert både gjennom Android sine hjemmesider, bøker og brukerforum. Det er også en gratis programvare som har åpen kildekode.



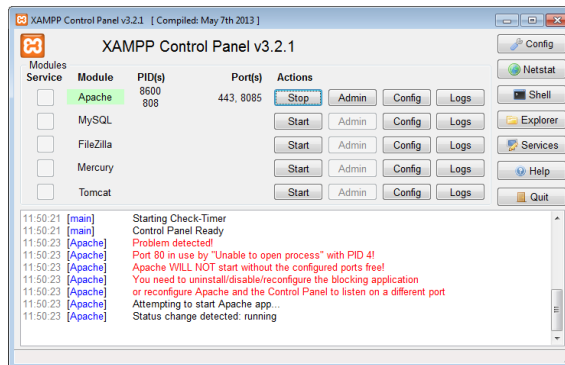
Figur 6.3: Eksempel på Android-programmering i Eclipse

⁵<http://www.eclipse.org/>

⁶<http://www.oracle.com/us/technologies/java/overview/index.html>

Før Android SDK ble valgt, ble et alternativ med et visuelt programmeringsspråk undersøkt. Med *Pure Data* har man gjennom programtillegget *PdDroidParty*⁷ mulighet for å kjøre *Pure Data*-programmer på Android, men grunnet manglende funksjonalitet og manglende muligheter for media-avspilling ble ikke denne programvaren valgt.

*Xampp*⁸ er et lite program som automatisk setter opp en *Apache*-tjener⁹ for brukeren. *Apache* er en åpen kildekode http-tjener utviklet av frivillige siden 1995, og har vært den mest populære web-tjeneren siden 1996¹⁰. Programmet er godt dokumentert på egen hjemmeside, og har et enkelt brukergrensesnitt. Figur 6.4 viser hvordan *Xampp* ser ut, og av figuren ser man at en *Apache*-tjener settes opp enkelt ved å trykke på en start-knapp.



Figur 6.4: Eksempel på oppsett av server med Xampp

⁷<http://droidparty.net/>

⁸<https://www.apachefriends.org/about.html>

⁹http://httpd.apache.org/ABOUT_APACHE.html

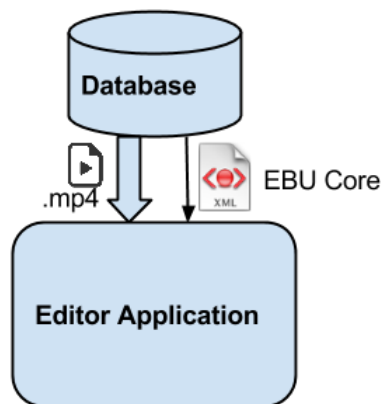
¹⁰http://httpd.apache.org/ABOUT_APACHE.html

6.5 Implementasjon av redaktørapplikasjon

For implementasjon av redaktørapplikasjonen ble programmet *Max 6* brukt. Her ble ikke hele applikasjonen utviklet fra bunnen av da en tidligere selvutviklet applikasjon egnet seg godt som et grunnlag. Dette var også fordelaktig i forhold til oppgavens omfang, og at man dermed fikk begrenset implementeringsmengden noe.

6.5.1 Import og database

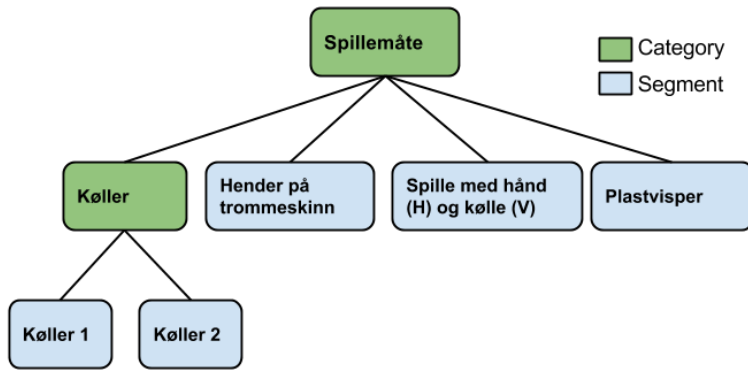
Første utfordring det ble arbeidet med var import av innhold til redaktørapplikasjonen. Figur 6.5 viser hvilken del av systemmodellen dette tilsvarer.



Figur 6.5: Import av innhold til redaktørapplikasjonen

Som et viktig første steg for å få redaktørapplikasjonen til å fungere etter kravene, ble det implementert en måte å automatisk kunne importere databasen. Dette er løst slik at brukeren kun skal legge inn en mappe med multimediaalt innhold, samt medfølgende metadatafiler, i applikasjonen. Resten

skjer automatisk. Metadatafilene er skrevet i XML, og følger EBU Core standarden. Et DOM objekt ble dermed brukt i Max for hente ut nødvendig data fra XML-filen slik at de kunne bli presentert i brukergrensesnittet til applikasjonen. Den mest kompliserte strukturen å importere har vært den hierarkiske strukturen av kategoriserte tagger knyttet til utdrag i en tidslinje. Siden dybden i strukturen kan variere, dvs. et utdrag kan høre til flere lag av kategorier og underkategorier, er dette løst ved å kun hente ut ett nivå av gangen der øverste nivå er det første som hentes ut.



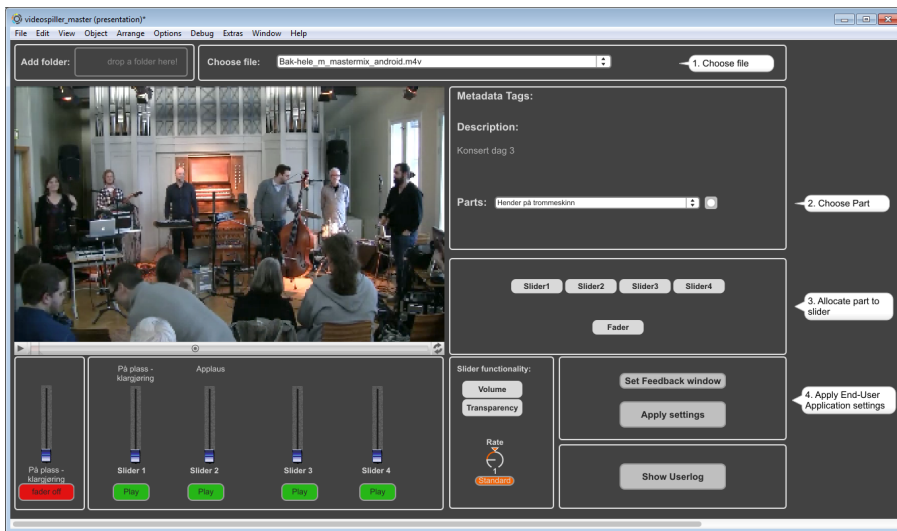
Figur 6.6: Utdrag av taggestruktur

For å vise et eksempel på denne strukturen er et utdrag av det taggete materialet som er importert i denne oppgaven, vist i figur 6.6. Av figuren ser man at foreldrenodene danner en kategoribeskrivelse, mens løvnødene er et tagget utdrag i tidslinjen.

6.5.2 Brukergrensesnitt og interaksjon

Siden Max baserer seg på visuell programmering, er det en veldig kort vei fra et implementert program til et brukergrensesnitt. Brukergrensesnitt utvikles i Max ved å inkludere ønskede deler av programmet til en presentasjonsmodus. Et ferdig program kan via innebygde funksjoner i Max eksporteres

som et “standalone”-program slik at programmet kan kjøres uavhengig av Max. For denne oppgaven ble det vanskelig å lage en universelt fungerende “standalone” grunnet at filbanen til filene tjeneren disponerer (og som redaktørapplikasjonen kommuniserer mot), vil endre seg fra maskin til maskin.



Figur 6.7: Hovedvindu i redaktørapplikasjon

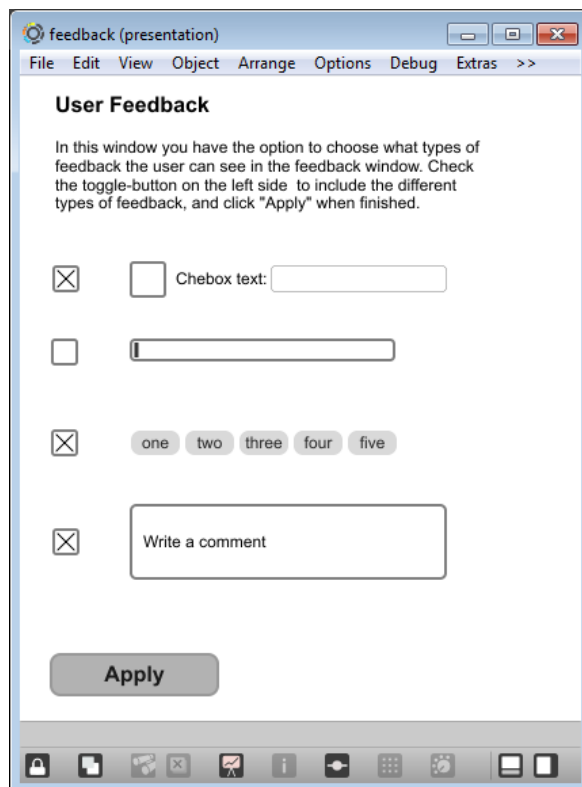
Figur 6.7 viser hovedvinduet til det endelige brukergrensesnittet for redaktørapplikasjonen. Av figuren ser man at brukeren har mulighet til å importere filinnholdet ved å dra en filmappe til ruten øverst i venstre hjørne. Rullegardinmenyen til høyre vil så vise alle multimediafiler, og ved valg av en fil vil man kunne se metadatainnholdet representert i ruten kalt “Metadata Tags”. Det som vises av metadata i denne ruten er en beskrivelse av filen samt en rullegardinmeny, kalt “Parts”, som viser taggestrukturen. Denne informasjonen hentes ut ved hjelp av et DOM objekt i Max. Siden utdrage- ne er kategorisert, og dermed danner en hierarkisk struktur, representeres hvert lag som listeelementer i rullegardinmenyen. Når brukeren så velger et element i rullegardinmenyen henter DOM objektet denne noden sine bar-

nenoder, som danner de nye elementene i menyen. Knappen til høyre for rullegardinmenyen gjør det mulig å gå tilbake til forrige nivå.

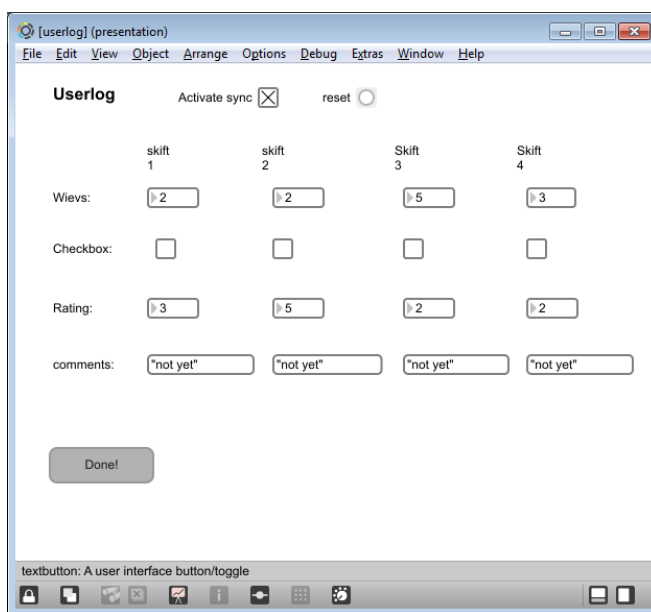
De taggete utdragene er mulig å spille av i applikasjonen ved å tilordne utdragene til glidebryterne vist nederst i figuren. Dette gjøres for eksempel ved å trykke på “slider 1”-knappen i den midterste ruten til høyre, for så å trykke på “play”-knappen til slideren for å spille av innholdet. Den første muligheten man har for å sette opp visningen i sluttbrukerapplikasjonen er ved å trykke på “Apply Settings”-knappen i nedre høyre hjørne av brukergrensesnittet. Da tilordnes elementene som er flyttet til glidebryterne til en liste som blir tilgjengelig for sluttbrukeren.

Figur 6.8 viser vinduet som åpner seg ved å trykke på “Set Feedback Window”-knappen i hovedvinduet. Her setter man opp hvilke typer tilbakemeldinger sluttbrukeren skal kunne gi i sluttbrukerapplikasjonen ved å trykke på avkrysningsboksene til venstre for tilbakemeldingstypen, og avslutte med “Apply”-knappen.

Ved å trykke på “Show Userlog”-knappen nederst til høyre i hovedvinduet åpnes et vindu som presenterer en brukerlogg, vist i figur 6.9. Av figuren ser man at man får en presentasjon av loggført informasjon fra brukeren. Loggen presenteres i en tabellstruktur der kolonnene sier hvilket utdrag som beskrives, og radene sier hva som er loggført. Ved å trykke på “Activate Sync”-avkrysningsboksen vil brukerloggen bli hentet ut fra tjeneren og vist i tabellstrukturen. Når brukeren trykker på ‘reset’-knappen vil brukerloggen nullstilles hos tjeneren.



Figur 6.8: Oppsett av tilbakemeldingsvindu



Figur 6.9: Vindu for visning av sluttbrukerlogg

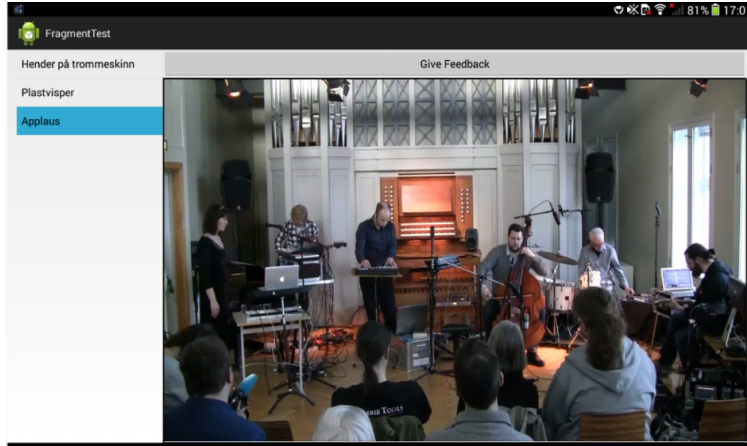
6.6 Implementasjon av sluttbrukerapplikasjon

Utviklingen av sluttbrukerapplikasjonen har blitt gjennomført ved hjelp av Android SDK der selve kodingen har foregått i Eclipse. For å spille av innholdet i applikasjonen ble mediefilene fra databasen manuelt plassert i en fast filbane på nettbrettet.

6.6.1 Brukergrensesnitt og interaksjon

Siden det i denne applikasjonen fra designprosessen, se figur 5.3 og 5.4, var ønsket et brukergrensesnitt der en liste alltid var synlig, er det tatt utgangspunkt i en mal i Eclipse kalt “Master detail flow”. I denne malen er det implementert en liste som lar navnet på et valgt listeelement bli presentert som en tekstboks i et detaljvindu til venstre for listen. Listervisning og detaljvindu er implementert med noe som kalles “fragmenter”¹¹. Et fragment er en aktivitet som kjøres på en ønsket del av skjermen, og som er tilpasset for å enkelt kunne bytte visning mellom andre fragmenter. Fragmentet for videovisning er dermed en modifisert versjon av fragmentet som viste en tekstboks i malen. Fragmentet for tilbakemelding er et nytt selvutviklet fragment.

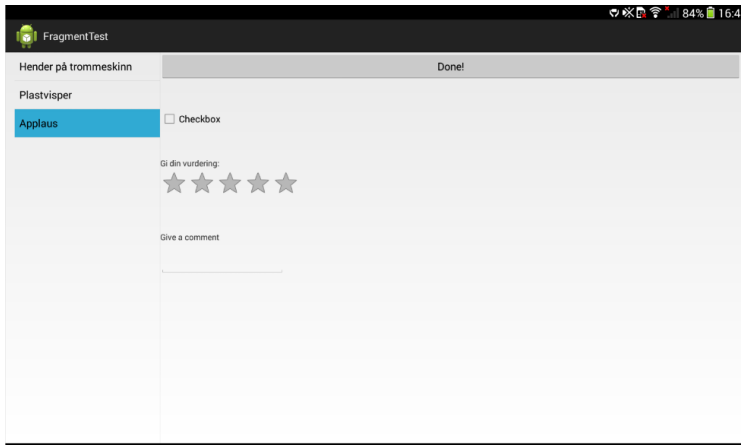
¹¹<http://developer.android.com/guide/components/fragments.html>



Figur 6.10: Brukergrensesnitt sluttbrukerapplikasjon: presentasjonsvindu

Det første vinduet som møter brukeren er vist i figur 6.10. I dette vinduet får brukeren innholdet presentert i en videovisning som startes ved at et element i listevisningen blir trykket på. Hva listen fylles med er på forhånd avgjort i redaktørapplikasjonen. Man har også mulighet til å starte/stoppe avspillingen i en mediekontroller som dukker opp ved berøring av avspilningsvinduet.

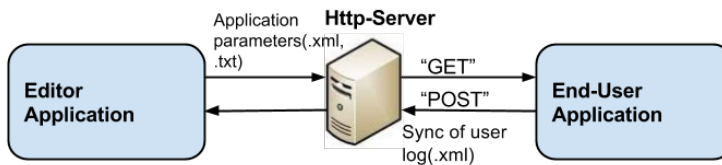
Når man trykker på “Give Feedback”-knappen over videovisningen (se figur 6.10), skifter visningen til tilbakemeldingsvinduet som er vist i figur 6.11. Her vises de tilbakemeldingstypene som redaktøren på forhånd har huket av. Brukeren kan etter avgitte tilbakemeldinger trykke på “Done”-knappen som gjør at vinduet skifter tilbake til avspillingen, eller velge et nytt utdrag for avspilling i listevisningen.



Figur 6.11: Brukergrensesnitt sluttbrukerapplikasjon; tilbakemeldingsvindu

6.7 Kommunikasjon og tjener

Denne delen forklarer hvordan data mellom redaktørapplikasjon og sluttbrukerapplikasjon har foregått. Siden det kun skulle utveksles enkle data mellom sluttbruker og redaktør, har det vært tilstrekkelig å bruke en http-tjener. Programmet Xampp ble brukt for å sette opp en denne tjeneren på samme maskin som redaktørapplikasjonen.



Figur 6.12: Modell for kommunikasjon via tjener

Figur 6.12 viser modellen for kommunikasjonen mellom applikasjonene. Siden tjeneren kjøres på samme maskin som redaktørapplikasjonen, lagres

konfigurasjonsfilene til den filbanen tjeneren deler på det lokale nettverket. Sluttbrukerapplikasjonen er dermed den eneste applikasjonen som henter og sender informasjon ved bruk av http-protokollen. Filene som benyttes i utvekslingen av data er; en XML-fil med parametere for tilbakemeldingstyper, en TXT-fil med data til listevisningen og en XML-fil med loggføring av aktivitet (se vedlegg A).

Sluttbrukerapplikasjonen henter ut dataene til disse filene gjennom to steg. Først sendes en "GET"-forespørsel mot tjeneren som gir en inputstrøm med data til sluttbrukerapplikasjonen. For å hente ut nødvendig informasjon fra inputstrømmen benyttes en DOM som det gjøres forespørsler mot. Det finnes kraftigere biblioteker for å gjøre mer spesifikke forespørsler mot et DOM-tre som "XPath"¹², men mer standard DOM-forespørsler holdt for de filene som har blitt brukt i denne oppgaven. For å kunne oppdatere brukerloggen blir det for hver hendelse sendt en "POST"-forespørsel mot tjeneren som oppdaterer XML-filen med brukerlogg.

¹²<http://www.w3.org/TR/xpath/>

Del III

Evaluering og konklusjon

Kapittel 7

Evaluering

I dette kapitlet gjennomføres en evaluering av det utviklede demo-systemet. Dette skjer i to steg. Først gjennom en brukertest av systemet som gir resultater som danner grunnlag for evaluering, og til slutt gjennom en diskusjon der systemet videre evalueres med fokus på nytteverdi og måloppnåelse.

7.1 Brukertest av demo-systemet

For å kunne evaluere nytten og brukbarheten til demo-systemet har det blitt gjennomført en brukertest. Testen har blitt gjennomført med en rekke testpersoner på sluttbrukersiden, og en forsker på redaktørsiden.

7.1.1 Forberedelser

I forkant av testen ble det utarbeidet en testinstruksjon (se vedlegg B.1) i samarbeid med redaktøren som under denne testen var Andreas Bergsland. Et krav var at testen skulle kunne settes opp i redaktørapplikasjonen og

at dataene skulle kunne hentes inn trådløst via det lokale nettverket slik systemet er utviklet. Det ble bestemt at innholdet som skulle brukes for testen er den samme som under utvikling (se åpningen av kapittel 6, “Valg av innhold”).

Testen ble på forhånd definert slik: En gruppe testpersoner skal innta rollen som sluttbrukere i demo-systemet. Testpersonene skal se gjennom fire utdrag fra musikkfremføringen, og for hvert utdrag observere (se og lytte) tydeligheten av skifter i musikken der musikken går fra ett uttrykk til et annet. For hvert utdrag skal testpersonen gi en tilbakemelding på hvor tydelig skiftet ble oppfattet i en likert-skala. Tilbakemeldingstypen som ble valgt i sluttbrukerapplikasjonen ble dermed tilbakemelingsvinduet “ratingbar” (se figur 6.11) bestående av fem stjerner der hver stjerne hadde sin gitte betydning opp mot likert-skalaen. Før testen ble gjennomført, ble det utført en pilot-test for å forsikre at alt fungerte som det skulle.

7.1.2 Gjennomføring

Totalt 20 personer gjennomførte testen som ble utført under kontrollerte omgivelser i et eget lab-rom. Testgruppen regnes som “ikke-profesjonelle” i den forstand at de ikke har tidligere erfaring med å vurdere musikk og ellers musikalske fremføringer.

Først ble testpersonen presentert med testinstruksjonen som de ble bedt om å lese gjennom. Deretter viste undertegnede kort applikasjonen og bisto med svar på eventuelle spørsmål. Testpersonen ble så plassert i et improvisert avlukke der han/hun fikk gjennomføre testen uten forstyrrelser. Samtidig satt undertegnede utenfor å observere at data ble samlet inn i redaktør-applikasjonen, og bisto hvis det skulle oppstå problemer underveis. Siden det ikke er implementert noen muligheter for å lage brukerprofiler, ble den genererte brukerlogg-filen for hver testgjennomføring lagret som ett resultat (se vedlegg B.2). I alt ble dermed 20 brukerlogg-filer generert. Figur 7.1 viser et bilde av en testgjennomføring der brukeren sitter i dette improviserte avlukket.



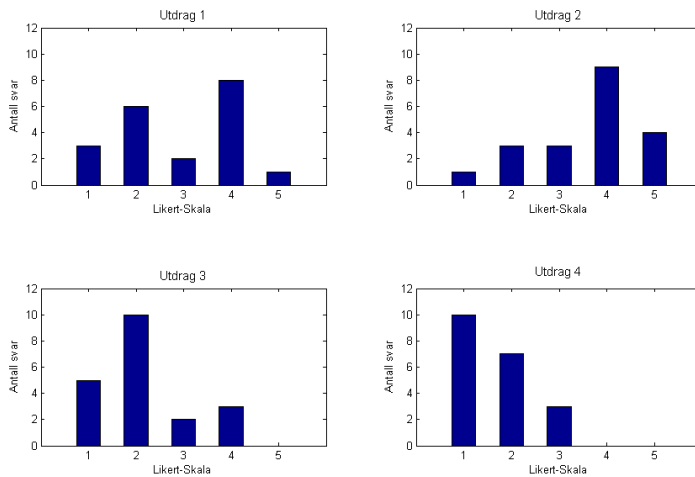
Figur 7.1: Brukertest gjennomføring

Til slutt ble testredaktør Andreas Bergsland presentert for resultatene for å gi tilbakemelding på resultatet og formeninger om systemet. Det er verdt å merke seg at testen gjerne skulle ha blitt gjennomført med hodetelefoner, men grunnet en defekt ved nettbrettet fungerte ikke hodetelefonutgangen.

7.1.3 Resultater og analyse

For å hente ut resultatene ble de genererte XML-filene med brukerlogg gjennomgått manuelt. De avgitte vurderingene ble manuelt ført inn i et “Matlab-

skript” (se vedlegg B.2) der resultatene ble samlet.



Figur 7.2: Resultat brukertest

Figur 7.2 viser en oversikt over resultatene i stolpediagram, der hvert diagram viser tilbakemeldingene fra ett enkelt utdrag. Likert-skalaen som danner x-aksen i resultatfiguren (figur 7.2) ble valgt som følger:

Likert-Skala:

- 1: Svært utydelig
- 2: Utydelig
- 3: Hverken tydelig eller utydelig
- 4: Tydelig
- 5: Svært tydelig

Ut fra figur 7.2 kan det se ut som gruppen med sluttbrukere har vært relativt

samstemte om at utdrag 3 og 4 inneholdt et mindre tydelig skifte, der utdrag 3 er minst tydelig. For utdrag 2 kan det se ut som det har vært en større enighet om et tydelig skifte, mens utdrag 1 fremstår som det mest tvetydige resultatet.

For videre analyse av resultatene er ITU sin teststandard ITU-T BT.500[5] brukt som utgangspunkt. Denne standarden tar egentlig for seg subjektive tester av bildekvalitet, noe som ikke er fokuset for denne testen. Men siden svarene avgis på en poengskala fra 1-5 kan samme tallanalyse av resultatene benyttes. Det er i henhold til standarden regnet på middelerverdi og et 95% konfidensintervall for utdragene.

Tabell 7.1: Analyse av resulater

	Middelerverdi	konfidensintervall
Utdrag 1	2.9	(2.4, 3.5)
Utdrag 2	3.6	(3.0, 4.2)
Utdrag 3	2.2	(1.6, 2.7)
Utdrag 4	1.7	(1.0, 2.3)

Ut fra tabell 7.1 ser man utregnet middelerverdi og konfidensintervall for de forskjellige utdragene. Av konfidensintervallene ser man at utdrag 2 og 4 ikke har overlappende intervaller og er dermed signifikant forskjellige. Dette stemmer godt overens med tidligere observasjoner.

Siden det er redaktøren Andreas Bergsland som har en interesse av resultatene fra brukertesten, har han blitt spurt om å gi sin mening på hvorvidt de innsamlede resultatene er interessante, og om de har noen verdi. Bergsland har svart følgende angående de genererte resultatene:

Testen gir et bilde av hvordan et sett med forsøkspersoner har respondert basert på avspilling av en video av en musikkframføring, og da med spesiell fokus på strukturelle skifter i musikken. Selv uten mer utførlige analyser av dette materialet, gir responsen et interessant bilde av hvordan noen passasjer framtrer med tydeligere skifter (udrag 2), mens andre oppfattes som mer uty-

delige (utdrag 3 og 4). Det er også interessant å se hvordan det virker å være større uenighet om utdrag 1. Hånes test kan gi utgangspunkt for en videre analyse av disse utdragene som kan gi et verdifullt bidrag til forskning rundt musikkutøvelse i frimprovisert ensemblemusikk.

7.2 Diskusjon

I denne delen vil resultatene av implementasjon og brukertest diskuteres med hovedfokus på måloppnåelse og nytteverdi. Det vil først fokuseres på demo-systemets nytteverdi for deretter å vurdere måloppnåelse opp mot kravspesifikasjonen. Til slutt vil forbedringspotensialet til systemet diskuteres.

7.2.1 Demo-systemets nytteverdi

Nytteverdien for dette systemet avgjøres på bakgrunn av hva brukerne av applikasjonene subjektivt mener om applikasjonene. Det er rimelig å anta at systemet er avhengig av et spennende innhold, med godt strukturert høynivå metadata, for at brukerne skal få best mulig følelse av nytte ved applikasjonene. Siden det ikke er gjennomført noen grundig test med intervjuer for å avdekke nytteverdien av sluttbrukerapplikasjonen er det vanskelig å trekke noen endelige konklusjoner om denne. Noe kan derimot sies om redaktørapplikasjonen på bakgrunn av Andreas Bergsland sine tilbakemeldinger. Det at Bergsland mente resultatene fra brukertesten var interessante og et godt utgangspunkt for videre forskning, er en god bekreftelse på at applikasjonen har en nytteverdi. En videre tilbakemelding fra Bergsland om systemet er som følger:

Systemets redaktørdel demonstrerer en interessant måte å vise og organisere materiale som er merket med metadata av forskeren, særlig fordi den også direkte kan brukes til å velge ut og

skreddersy et oppsett for å innhente respons fra brukere av den mobile applikasjonen gjennom disses tilknytning til en felles server. Dette kan gjøre det enkelt for forskeren å innhente et stort responsmateriale, med tanke på utbredelsen av ulike typer mobile plattformer og enheter vi ser i dag.

Denne tilbakemeldingen bidrar også positivt for systemet nytteverdi. For å kunne avgjøre nytteverdien mer sikkert vil man trenge flere forskere som får testet applikasjonen med sitt innhold, og gitt sine vurderinger. Dette vil mest sannsynlig også kunne avdekke en del feil og mangler ved applikasjonen, og demo-systemet generelt.

7.2.2 Validering av kravoppnåelse

Det vil her gås gjennom hvorvidt forventninger og krav til demo-systemet har blitt tilfredsstilt. Noen av kravene er ikke veldig spesifikke, og det kan være vanskelig å vurdere i hvilken grad de er oppnådd. For redaktørapplikasjonen er alle de funksjonelle kravene oppnådd i større eller mindre grad. For krav 3 om å kunne styre en liste hos sluttbruker er det ganske åpenbart at det kunne vært praktisk å tildele flere enn fire elementer til listen. Funksjonelt krav 5 om å visuelt fremstille en loggføring av data er også oppnådd, men opprettelsen av brukerprofiler på sluttbrukersiden kunne ledet til mer interessant analyse av tilbakemeldinger med f.eks. sammenligningsmuligheter og visning av akkumulerte data som i resultatdelen av brukertesten(se figur 7.2).

For sluttbrukerapplikasjonen er det mer klart at noen krav ikke er nådd. Funksjonelt krav 4 om muligheter for passiv tilbakemelding samt krav 5 om fullskjermmodus, ble ikke møtt grunnet nedprioriteringer knyttet til omfang og tid tilgjengelig. Muligheter for passiv tilbakemelding hadde åpnet for en interessant analyse av systemet da det hadde gitt flere tilbakemeldingsmuligheter sammenlignet med tradisjonell datainnhenting. Fullskjermmodus ble forsøkt implementert, men dette ledet ikke til noen godt fungerende løsning.

Det er tydelig at en spissere definering av kravspesifikasjonen der suksesskriteriene for kravene var mer objektivt målbare ville gjort valideringen av funksjonell kravspesifikasjon bedre. Da det ikke er gjennomført noen brukbarhetstest av hverken redaktørapplikasjonen eller sluttbrukerapplikasjonen er det vanskelig å bedømme med sikkerhet om de ikke-funksjonelle kravene er nådd. Likevel er det verdt å nevne at testpersonene involvert i brukertesten virket å oppleve sluttbrukerapplikasjonen som enkel å navigere seg rundt i, og forstå. Utvikleren selv mener at responstiden ved interaksjon er god for begge applikasjonene i henhold til ikke-funksjonelt krav 2 for begge applikasjonene.

7.2.3 Kritikk til demo-system

Selv om dette systemet oppfyller de fleste av de på forhånd stilte krav, er det er også forhold som gjør at man kanskje vil stille seg noe kritisk til demo-systemet. Innenfor forskning stilles det ofte strenge krav til at brukertester skal gjennomføres under kontrollerte forhold. Ut fra konseptet skal dette systemet kunne anvendes i en UMA sammenheng, der alle skal få tilgang, noe som ikke leder til veldig kontrollerte forhold under innsamling av data. Med dette systemet er det flere aspekter som gjør at dataene som samles inn kan miste sin integritet. Sluttbrukeren kan oppholde seg i miljø som gjør at ytre forhold svekker konsumeringen av innholdet, som for eksempel lysforhold og støy. Også tekniske forhold som skjermstørrelse og lyd fra høyttalere/hodetelefoner kan være kilder til usikre data. Nettbrettet til sluttbrukerapplikasjonen har en skjermstørrelse på 10 tommer noe som ikke er en liten skjermstørrelse for et nettbrett, men det kan være en begrensning for en redaktør som ønsker at sluttbrukeren skal kunne oppfatte og vurdere noe visuelt ved innholdet. Det at demo-systemet ikke har noen fungerende fullskjermmodus er også en ulempe i denne sammenheng. Begrensningen blir helt klart større for smart-telefoner der skjermstørrelsen er rundt 5 tommer på det beste. Lyd fra høyttalerne til et nettbrett er sjelden god, så ofte vil det være fordelaktig at sluttbrukeren bruker hodetelefoner som gir et bedre lydbilde, og samtidig skjermer brukeren fra omgivelsene. Tiltak som

kan innføres for å bedre ytre forhold og tekniske utfordringer kan være å ikke tillate avspilling før hodetelefoner er tilkoblet, sette nedre grense for skjermstørrelse eller å gi redaktøren muligheter til å anbefale konsumeringsforhold til sluttbrukeren i form av en dialogboks eller lignende. Alle disse tiltakene for å optimalisere presentasjon hos sluttbruker har sin pris i forhold til utstrekning der antall brukere som kan konsumere innholdet vil bli begrenset.

For redaktørapplikasjonen er det som nevnt i valideringen en ganske stor begrensning at det kun kan legges til fire elementer i listen som gjøres tilgjengelig for sluttbruker. Dette skyldes at applikasjonen som ble brukt som utgangspunkt for implementering av redaktørapplikasjonen hadde muligheter for tilordning av utdrag til fire glidebrytere, og var dermed praktisk å bruke som utgangspunkt for tilordning av listeelementer til sluttbrukeren. For å utbedre dette kan det implementeres muligheter for å tilordne utdrag i en listevissning eller rullegardinmeny som gjør at listelengde ikke vil ta opp noe større eller mindre plass på skjermen. Dette kan likevel ha en innvirkning på brukergrensesnittets oversiktighet, og må derfor tas hensyn til ved implementering.

Slik det nå blir samlet inn data fra sluttbruker til redaktør er det ingen mulighet for å integrere brukerloggen opp mot metadatafilen som beskriver innholdet. Brukerloggen ligger riktignok lagret i en egen fil, og man har mulighet til å få informasjonen visualisert i applikasjonen, men det vil trolig skape en større verdi om informasjonen kan integreres mot innholdets metadatafiler. EBU Core standarden åpner for temmelig rik annotasjon med muligheter for å tilordne kommentarer og “rating” for en hel fil, så det er rimelig å anta at denne funksjonen er gjennomførbar å implementere. Integrering av brukerlogg gir også verdi i form av at dataene kan eksporteres i et standardisert format, noe som bidrar til interoperabilitet.

Kapittel 8

Konklusjon

Det har i denne oppgaven blitt utviklet et multiplattform demo-system som demonstrerer to applikasjoner med forskjellige agendaer og interaksjonsmuligheter som opptrer i et trådløst nettverk. Hovedfokuset har ligget på redaktørapplikasjonen og konfigurasjonsmulighetene som åpner seg ved innføring av en slik nyskapende applikasjon. Systemet mottok positive tilbakemeldinger fra Andreas Bergsland som har deltatt på en brukertest av demo-systemet. Gjennom Bergsland sine tilbakemeldinger har man avdekket at det kan være et behov for et slikt system, og at systemet innfører en interessant måte å videre berike multimedialt innhold med metadata.

De fleste krav til systemet ble oppnådd, så implementasjonsmessig har man grunn til å være fornøyd med resultatet. Likevel er det bekymringer knyttet til dette konseptet. Primært dreier disse seg om at datainnhentingene beveger seg fra en tradisjonell metode med kontrollerte omgivelser der man har god erfaring med dataenes integritet, og over til denne formen der videre forskning vil kreves for å kunne avgjøre verdien og integriteten til dataene.

Dette systemet har foreløpig kun blitt testet med én type database og i en forskningssammenheng. Likevel kan man se for seg at demo-systemet kan

tilpasses for å tjene en mer generell funksjon. Ved å inkludere flere former for metadata i systemet kan det godt tenkes at personer med andre agendaer og motiver kan ha nytte av dette systemet. Dette området kan eksempelvis være e-læring, der en sluttbruker kan ha utbytte av å kunne la metadata berike brukeropplevelsen av en forelesning. Redaktøren på sin side kan være en foreleser, og kan ha utbytte av tilbakemeldinger fra en gruppe sluttbrukere. En mer generell bruk av systemet kan oppnås ved å tillate en mer dynamisk tilpasning av brukergrensesnittet til redaktørapplikasjonen der redaktøren selv kan bestemme hvilken type metadata som skal visualiseres. Dette kan være praktisk da det er rimelig å anta at for mye presentasjon av forskjellig typer metadata vil skape et kaotisk og uoversiktlig brukergrensesnitt.

8.1 Videre utvikling

Som en videre utvikling av dette systemet er det naturlig å jobbe videre mot å oppnå målene i kravspesifikasjonen, samt å utbedre de kravene som har fått sine bemerkninger. For å gjøre systemet mer tilgjengelig kan det også være interessant å åpne for muligheten til at sluttbruker- og kanskje også redaktørapplikasjonen gjøres tilgjengelig på flere enheter. For å bedre kartlegge integriteten til dataene som genereres, kan en videre utvikling være å undersøke påvirkningen av innsamlede data ved ukontrollerte avspilningsforhold. Innføring av passive tilbakemeldingsmuligheter som å bruke bevegelses-sensoren i smarte enheter, og ellers annen sensorteknologi kan være interessante områder for videre utvikling av konseptet.

Bibliografi

- [1] Cisco, 2014, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013?2018”
- [2] Briony J Oates, 2006, “Researching Information Systems and Computing”
- [3] Fernando Pereira, Ian Burnett, 2003, “Universal Multimedia Experiences for Tomorrow”
- [4] Fernando Kuipers, Robert Kooij, Danny De Vleeschauwer, and Kjell Brunnström, 2010, “Techniques for Measuring Quality of Experience”
- [5] ITU , 2012, “Recommendation ITU-R BT.500-13 Methodology for the subjective assessment of the quality” of television pictures
- [6] TECH 3293 EBU CORE METADATA SET v.1.5, EBU operating eurovision and euroradio
- [7] Jean-Pierre EVAIN, 2013, “EBUCore 2013”
- [8] J. Kunze, T. Baker, 2007, “The Dublin Core Metadata Element Set”, <http://www.ietf.org/rfc/rfc5013.txt>
- [9] Gilliland A.J, 2005, “Introduction to metadata: Pathways to digital information”
- [10] Bhavani Thuraisingham, 2001, “Managing annd Mining Multimedia Databases”
- [11] BERNHARD HASLHOFER and WOLFGANG KLAS, 2010, “A Survey of Techniques for Achieving Metadata Interoperability”
- [12] NISO Press, 2004, “Understanding Metadata”
- [13] Tim Bray Jean Paoli, C.M. Sperberg-McQueen, Eve Maler, Francois Yergeau, 2008, “Jean Paoli,Extensible Markup Language (XML) 1.0 (Fifth Edition)”, <http://www.w3.org/TR/xml/>
- [14] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee , 1999,“Hypertext Transfer Protocol – HTTP/1.1”

- [15] Darcy Haag Granello and Joe E. Wheaton, 2004, "Online Data Collection: Strategies for Research"
- [16] Dane Bertram, 2007, "Likert Scales" , <http://poincare.matf.bg.ac.rs/~kristina/topic-dane-likert.pdf>
- [17] Khalil ur Rehman Laghari, Noel Crespi, B. Molina, C.E. Palau , 2011, "QoE aware Service Delivery in Distributed Environment"
- [18] Markus Waltl, Christian Timmerer, Hermann Hellwagner, 2009, "A Test-Bed for Quality of Multimedia Experience Evaluation of Sensory Effects"
- [19] Janne Seppänen, Martín Varela, Aggeliki Sgora, 2013, "An autonomous QoE-driven network management framework"
- [20] Sujeet Mate, Umesh Chandra, Igor D.D. Curcio ,2006, "Movable-Multimedia: Session Mobility in Ubiquitous Computing Ecosystem"
- [21] R. Shacham, H. Schulzrinne, S. Thakolsri, W. Kellerer, 2009, "Session Initiation Protocol (SIP) Session Mobility", <https://tools.ietf.org/html/rfc5631>
- [22] Ricardo Mendes Costa Segundo, Celso Alberto Saibel Santos, 2013, "Second Screen Event Flow Synchronization"
- [23] Francesco Ricci, Lior Rokach, Bracha Shapira ,2011, Chapter 1 " Introduction to Recommender Systems Handbook" in "Recommender Systems Handbook"
- [24] Jia-Ching Ying, Huan-Sheng Chen, Kawuu W. Lin, Eric Hsueh-Chan Lu, Vincent S. Tseng, Huan-Wen Tsai, Kuang Hung Cheng, Shun-Chieh Lin, 2014, "Semantic trajectory-based high utility item recommendation system"
- [25] Joel R. Evans and Anil Mathur , 2005, "The value of online surveys"

Vedlegg A

Tjener-filer

I denne delen presenteres filene som ble brukt for utveksling av informasjon hos tjeneren.

Under vises filen ble brukt for å tilordne initiell informasjon om hvilken tilbakemeldingsmetode som kunne gis i sluttbrukerapplikasjonen:

initial.xml:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <initial_settings>
3   <toggle>1</toggle>
4   <toggletxt>Checkbox</toggletxt>
5   <rating>1</rating>
6   <comment>1</comment>
7 </initial_settings>
```

Under vises et eksempel på filen som ble brukt for å hente inn informasjon om listeelementene:

segmentList.txt:

```
1 1, Hender på trommeskinn, Bak-hele_m_mastermix_android.m4v,
   510000, 600000;
2 2, Plastvisper, Bak-hele_m_mastermix_android.m4v, 968000,
   1594000;
```

```
3 | 3, Applaus, Bak-hele_m_mastermix_android.m4v, 3450000, 3492000;
```

Under vises koden til XML-filen som samler brukerlogg:

testlist.xml:

```
1 | <?xml version="1.0" encoding="UTF-8"?><userinfo>
2 |   <item id="1">
3 |     <count>2</count>
4 |     <toggle>0</toggle>
5 |     <toggletxt>not yet</toggletxt>
6 |     <rating>3.0</rating>
7 |     <comment>symbol not yet</comment>
8 |   </item>
9 |   <item id="2">
10 |     <count>4</count>
11 |     <toggle>1</toggle>
12 |     <toggletxt>not yet</toggletxt>
13 |     <rating>5.0</rating>
14 |     <comment>symbol dette var knall</comment>
15 |   </item>
16 |   <item id="3">
17 |     <count>7</count>
18 |     <toggle>0</toggle>
19 |     <toggletxt>not yet</toggletxt>
20 |     <rating>2.0</rating>
21 |     <comment>symbol not yet</comment>
22 |   </item>
23 |   <item id="4">
24 |     <count>3</count>
25 |     <toggle>1</toggle>
26 |     <toggletxt>not yet</toggletxt>
27 |     <rating>2.0</rating>
28 |     <comment>symbol not yet</comment>
29 |   </item>
30 | </userinfo>
```

Vedlegg B

Brukertest

B.1 Testinstruksjon

Under vises Testinstruksjonen som ble brukt under testen:

testinstruks.rtf:

Testinstruksjon

Du vil nå få presentert 4 utdrag fra en konsert med musikkgruppen T-EMP. Alle utdragene viser skifter i musikken, dvs. der musikken går fra én type uttrykk til et annet. Din oppgave er å vurdere hvor tydelig du synes disse skiftene er. Hvert utdrag er merket som et listeelement i applikasjonen, og startes ved å trykke på det gitte elementet. Utdraget er ferdig når avspillingen stopper, og etter Utdraget er startet skal det ikke stoppes manuelt. Se gjennom alle utdrag så mange ganger du ønsker, og for hvert utdrag som har blitt sett, gi din vurdering ved å trykke på "Give Feedback"- knappen. I tilbakemeldingsvinduet skal du gi en bedømmelse på en skala fra 1-5 stjerner der antall stjerner betegner: Skala:

1 stjerne: Svært utydelig 2 stjerner: Utydelig 3 stjerner: Hverken tydelig eller utydelig 4 stjerner: Tydelig 5 stjerner: Svært tydelig

Du kan avgi svar så mange ganger du ønsker på hvert utdrag, der siste avgitte svar blir den stående vurderingen.

B.2 Kildekode fra brukertest

Eksempel på innhentet brukerlogg fra brukertesten:

kandidatXX.xml:

```

1 <?xml version="1.0" encoding="UTF-8"?><userinfo>
2   <item id="1">
3     <count>1</count>
4     <toggle>0</toggle>
5     <toggletxt>not yet</toggletxt>
6     <rating>4.0</rating>
7     <comment>symbol not yet</comment>
8   </item>
9   <item id="2">
10    <count>1</count>
11    <toggle>0</toggle>
12    <toggletxt>not yet</toggletxt>
13    <rating>3.0</rating>
14    <comment>symbol not yet</comment>
15  </item>
16  <item id="3">
17    <count>1</count>
18    <toggle>0</toggle>
19    <toggletxt>not yet</toggletxt>
20    <rating>2.0</rating>
21    <comment>symbol not yet</comment>
22  </item>
23  <item id="4">
24    <count>1</count>
25    <toggle>0</toggle>
26    <toggletxt>not yet</toggletxt>
27    <rating>1.0</rating>
28    <comment>symbol not yet</comment>
29  </item>
30 </userinfo>

```

Matlab-skript brukt for å samle testdata, samt å genere resultatplot:

testplot.m:

```

1 feedback1 = [4 2 5 4 4 4 2 1 3 4 4 2 1 2 4 4 1 2 2 3];

```

```
2 feedback2 = [4 4 5 2 4 2 4 2 4 4 3 3 1 3 5 5 4 4 4 5];
3 feedback3 = [2 1 3 3 2 1 1 2 2 2 2 2 1 4 4 4 2 2 1 2];
4 feedback4 = [2 1 2 2 1 1 1 3 1 2 1 3 1 1 3 2 1 2 1 2];
5
6 resultat1 = resultCounter(feedback1);
7 resultat2 = resultCounter(feedback2);
8 resultat3 = resultCounter(feedback3);
9 resultat4 = resultCounter(feedback4);
10
11 stars = [1 2 3 4 5];
12
13 figure
14 subplot(2,2,1);
15 bar(stars, resultat1, 0.5);
16 title('Utdrag 1')
17 xlabel('Likert-Skala')
18 ylabel('Antall svar')
19
20 subplot(2,2,2);
21 bar(stars, resultat2, 0.5);
22 title('Utdrag 2')
23 xlabel('Likert-Skala')
24 ylabel('Antall svar')
25
26 subplot(2,2,3)
27 bar(stars, resultat3, 0.5)
28 title('Utdrag 3')
29 xlabel('Likert-Skala')
30 ylabel('Antall svar')
31
32 subplot(2,2,4)
33 bar(stars, resultat4, 0.5)
34 title('Utdrag 4')
35 xlabel('Likert-Skala')
36 ylabel('Antall svar')
```